6.23 Subtlies of Coding

In the Huffman code, the bit sequences that represent individual symbols can have differing lengths so the bitstream index m does not increase in lock step with the symbol-valued signal's index *n*. To capture how often bits must be transmitted to keep up with the source's production of symbols, we can only compute averages. If our source code averages

$$B\overline{(}A)$$

bits/symbol and symbols are produced at a rate R, the average bit rate equals

$$B\overline{(}A)R$$

, and this quantity determines the bit interval duration *T*.

Exercise 6.23.1

Calculate what the relation between *T* and the average bit rate

$$B\overline{(}A)R$$

is.

A subtlety of source coding is whether we need "commas" in the bitstream. When we use an unequal number of bits to represent symbols, how does the receiver determine when symbols begin and end? If you created a source code that **required** a separation marker in the bitstream between symbols, it would be very inefficient since you are essentially requiring an extra symbol in the transmission stream.

NOTE: A good example of this need is the Morse Code: Between each letter, the telegrapher needs to insert a pause to inform the receiver when letter boundaries occur.

As shown in this example (Compression and the Huffman Code (Page 297)), no commas are placed in the bitstream, but you can unambiguously decode the sequence of symbols from the bitstream. Huffman showed that his (maximally efficient) code had the **prefix** property: No code for a symbol began another symbol's code. Once you have the prefix property, the bitstream is partially self-synchronizing: Once the receiver knows where the bitstream starts, we can assign a unique and correct symbol sequence to the bitstream.

Exercise 6.23.2

Sketch an argument that prefx coding, whether derived from a Hufman code or not, will provide unique decoding when an unequal number of bits/symbol are used in the code.

However, having a prefx code does not guarantee total synchronization: After hopping into the middle of a bitstream, can we always find the correct symbol boundaries? The self-synchronization issue does mitigate the use of efcient source coding algorithms.

Exercise 6.23.3

Show by example that a bitstream produced by a Hufman code is not necessarily self-synchronizing. Are fxed-length codes self synchronizing?